**Grammar G (After eliminating of left recursion and common left prefixes)**

Legend

**Green** = Same as original.

**Red** = Original row that need to be changed.

**Blue** = Result of the red row after change.

**\*** **type\_name** token is actually same as **id** token.

PROGRAM 🡪 BLOCK

BLOCK 🡪 **block** DEFINITIONS ; **begin** COMMANDS ; **end**

DEFINITIONS 🡪 DEFINITION | DEFINITIONS ; DEFINITION

DEFINITIONS 🡪 DEFINITION DEFINITIONS\_TAG

DEFINITIONS\_TAG 🡪

DEFINITIONS\_TAG 🡪 ; DEFINITION DEFINITIONS\_TAG

DEFINITION 🡪 VAR\_DEFINITION

DEFINITION 🡪 TYPE\_DEFINITION

VAR\_DEFINITION 🡪 id : BASIC\_TYPE | id : type\_name

VAR\_DEFINITION 🡪 id : VAR\_DEFINITION\_TAG

VAR\_DEFINITION\_TAG 🡪 BASIC\_TYPE

VAR\_DEFINITION\_TAG 🡪 type\_name

TYPE\_DEFINITION 🡪 **type** type\_name **is** TYPE\_INDICATOR

TYPE\_INDICATOR 🡪 BASIC\_TYPE

TYPE\_INDICATOR 🡪 ARRAY\_TYPE

TYPE\_INDICATOR 🡪 POINTER\_TYPE

BASIC\_TYPE 🡪 **integer**

BASIC\_TYPE 🡪 **real**

ARRAY\_TYPE 🡪 **array** [ SIZE ] **of** BASIC\_TYPE

POINTER\_TYPE 🡪 ^ BASIC\_TYPE | ^ type\_name

POINTER\_TYPE 🡪 ^ POINTER\_TYPE\_TAG

POINTER\_TYPE\_TAG 🡪 BASIC\_TYPE

POINTER\_TYPE\_TAG 🡪 type\_name

SIZE 🡪 int\_num

COMMANDS 🡪 COMMAND | COMMANDS ; COMMAND

COMMANDS 🡪 COMMAND COMMANDS\_TAG

COMMANDS\_TAG 🡪

COMMANDS\_TAG 🡪 ; COMMAND COMMANDS\_TAG

COMMAND 🡪 RECEIVER = EXPRESSION

COMMAND 🡪 **when** ( EXPRESSION rel\_op EXPRESSION ) **do** COMMANDS ; **default** COMMANDS ; **end\_when**

COMMAND 🡪 **for** ( id = EXPRESSION ; id rel\_op EXPRESSION ; id ++ ) COMMANDS ; **end\_for**

COMMAND 🡪 id = **malloc** ( **size\_of** ( type\_name ) )

COMMAND 🡪 **free** ( id )

COMMAND 🡪 BLOCK

RECEIVER 🡪 id | id [ EXPRESSION ] | id ^

RECEIVER 🡪 id RECEIVER\_TAG

RECEIVER\_TAG 🡪

RECEIVER\_TAG 🡪 [ EXPRESSION ]

RECEIVER\_TAG 🡪 ^

EXPRESSION 🡪 int\_num | real\_num | id | id [ EXPRESSION ] | & id | id ^ | **size\_of** ( type\_name ) | id ar\_op EXPRESSION

EXPRESSION 🡪 int\_num

EXPRESSION 🡪 real\_num

EXPRESSION 🡪 & id

EXPRESSION 🡪 **size\_of** ( type\_name )

EXPRESSION 🡪 id EXPRESSION\_TAG

EXPRESSION\_TAG 🡪

EXPRESSION\_TAG 🡪 [ EXPRESSION ]

EXPRESSION\_TAG 🡪 ^

EXPRESSION\_TAG 🡪 ar\_op EXPRESSION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Variables | Nullable | First | Follow |
| 1 | PROGRAM | - | { block } | { EOF } |
| 2 | BLOCK | - | { block } | { EOF, ;} |
| 3 | DEFINITIONS | - | { id, type } | { ; } |
| 4 | DEFINITIONS\_TAG | yes | { ; } | { ; } |
| 5 | DEFINITION | - | { id, type } | { ; } |
| 6 | VAR\_DEFINITION | - | { id } | { ; } |
| 7 | VAR\_DEFINITION\_TAG | - | { integer, real, type\_name } | { ; } |
| 8 | TYPE\_DEFINITION | - | { type } | { ; } |
| 9 | TYPE\_INDICATOR | - | { integer, real, array, ^ } | { ; } |
| 10 | BASIC\_TYPE | - | { integer, real } | { ; } |
| 11 | ARRAY\_TYPE | - | { array } | { ; } |
| 12 | POINTER\_TYPE | - | { ^ } | { ; } |
| 13 | POINTER\_TYPE\_TAG | - | { integer, real, type\_name } | { ; } |
| 14 | SIZE | - | { int\_num } | { ] } |
| 15 | COMMANDS | - | { id, when, for, free, block } | { ; } |
| 16 | COMMANDS\_TAG | yes | { ; } | { ; } |
| 17 | COMMAND | - | { id, when, for, free, block } | { ; } |
| 18 | RECEIVER | - | { id } | { = } |
| 19 | RECEIVER\_TAG | yes | { [, ^ } | { = } |
| 20 | EXPRESSION | - | { int\_num, real\_num, &, size\_of, id } | { ;, rel\_op, ), ] } |
| 21 | EXPRESSION\_TAG | yes | { [, ^, ar\_op } | { ;, rel\_op, ), ] } |